9-1996

An on-the-fly decoding technique for Reed-Solomon codes

Yuan Xing LEE
National University of Singapore

Robert H. DENG
Singapore Management University, robertdeng@smu.edu.sg

Eng Hean KOH
Seagate Technology

DOI: https://doi.org/10.1109/20.539231

Published in IEEE Transactions on Magnetics, 1996 September, 32 (5): 3962-3964.
https://doi.org/10.1109/20.539231

An on-the-fly Decoding Technique for Reed-Solomon Codes

Yuan Xing Lee[1], Robert H. Deng[2], Eng Hean Koh[3]
1. Magnetics Technology Centre, NUS, 10 Kent Ridge Crescent, Singapore 119260
2. Institute of Systems Science, NUS, 10 Kent Ridge Crescent, Singapore 119260
3. Seagate Technology International, Science Park Drive, #02-63, Singapore119260

Abstract ----**An on-the-fly error correction technique for double-byte-error-correction (DBEC) and triple-byte-error-detection (TBED) Reed-Solomon (RS) codes is presented in this paper. This new algorithm finds number of byte-errors (single byte-error, double-byte errors, and triple-byte errors) by simply testing the values of a few syndrome functions, and locates and corrects the byte-errors directly without using the standard iterative algorithms for finding the error location polynomial. More importantly, it neither suffers from malfunctions incurred in Deng-costello algorithm [1], nor requires syndrome re-calculation as in Koksal-Yucel's modification [2]. It is also much more simpler and faster than the original Deng-Costello algorithm. It has found applications in high-end disk drives where powerful on-the-fly correction is necessary.**

I. Introduction

The recording density of disk drives is going up steadily at a compound rate of 60% annually. This increase in density makes channels run faster and tend to be corrupted by more errors. To deal with this type of channels, powerful error correction is needed and must be done on-the-fly. It is known that Deng and Costello proposed a fast decoding algorithm for DBEC-TBED RS codes[1], which was shown later by Koksal and Yucel in [2] to be a few times faster than the standard Berlekamp-Massey decoding algorithm, but malfunctions in detection of certain triple-byte errors. An additional syndrome re-calculation needs to be added to Deng-Costello algorithm to overcome the malfunctions, thus slows down the speed of decoding [2].This paper presents a new approach to the decoding of DBEC-TBED RS codes, which is shown to be much more efficient than Deng-Costello algorithm without having any malfunctions. It is suitable for hardware implementation of on-the-fly correction in today's high-end disk drives.

II. Derivation of the Decoding Algorithm

Assume

$$g(x)=(x+\alpha^{m_0})(x+\alpha^{m_0+1})(x+\alpha^{m_0+2})(x+\alpha^{m_0+3})(x+\alpha^{m_0+4}) = g_0 + g_1x + g_2x^2 + g_3x^3 + g_4x^4 + g_5x^5 \quad (1)$$

Manuscript received Dec.1, 1995. YX Lee: dsiliyx@nus.sg.
* Project granted by Seagate Technology International.

is a generator polynomial for DBEC-TBED RS codes, where α is a primitive element of GF(2^m) and m_0 is any integer. The simplest encoder is obtained by setting $m_0 = -2$, since then $g_0 = g_5 = 1$, $g_1 = g_4$, $g_2 = g_3$. The code length $n \le 2^m - 1$, the number of check symbols in every code word is 5, and the dimension of the code k = n-5. This code has a minimum Hamming distance 6; therefore, it is capable of correcting all the single and double byte-errors per received word and simultaneously detecting all the triple byte-errors per received word.

Suppose that a code word is transmitted through a noisy channel, and its received word has v byte errors at locations $0 \le j_1 < j_2 < \cdots < j_v \le n-1$. Then the error polynomial is defined by

$$e(x) = Y_1x^{j_1} + Y_2x^{j_2} + \cdots + Y_vx^{j_v} \quad (2)$$

where Y_i denotes the error value at location j_i. Define

$$X_i = \alpha^{j_i}, \qquad i = 1, 2, ..., v \quad (3)$$

as error-location numbers. If X_i is known, the error location j_i of the i-th byte-error is also known. The syndrome of the received word may be computed from the various known methods with syndrome values given by

$$S_j = \sum_{i=1}^{v} Y_i X_i^{m_0+j}, \quad v = 1,2,.... ; j = 0,1,2,3,4 \quad (4)$$

If no error has occurred in the received word, then

$$S_j = 0, \qquad j = 0, 1, 2, 3, 4. \quad (5)$$

Define error location polynomial as

$$\sigma(x) = (x + X_1)(x + X_2)....(x + X_v) = x^v + \sigma_1x^{v-1} + ... + \sigma_{v-1}x + \sigma_v \quad (6)$$

From [3], we have the following equation:

$$S_j\sigma_v + S_{j+1}\sigma_{v-1} + ... + S_{j+v-1}\sigma_1 + S_{j+v} = 0, \quad 0 \le j \le 4 - v \quad (7)$$

In the case of v = 1, i.e., there is a single byte-error in the received word, equation (7) becomes

$$\begin{cases} S_0\sigma_1 + S_1 = 0 & (8.1) \\ S_1\sigma_1 + S_2 = 0 & (8.2) \\ S_2\sigma_1 + S_3 = 0 & (83) \\ S_3\sigma_1 + S_4 = 0 & (8.4) \end{cases}$$

and equation (6) reduces to

$$\sigma(x) = x + \sigma_1 \quad (9)$$

The coefficient σ_1 can be determined from equations (8.1) to (8.4) as

$$\sigma_1 = \frac{S_1}{S_0} = \frac{S_2}{S_1} = \frac{S_3}{S_2} = \frac{S_4}{S_3} \qquad (10)$$

Define

$$\beta_0 = S_1^2 + S_0 S_2 \qquad (11.1)$$

$$\beta_1 = S_2^2 + S_1 S_3 \qquad (11.2)$$

$$\beta_2 = S_1 S_4 + S_2 S_3 \qquad (11.3)$$

$$\beta_3 = S_3^2 + S_2 S_4 \qquad (11.4)$$

$$\beta_4 = S_0 S_3 + S_1 S_2 . \qquad (11.5)$$

Since v = 1, from (4) we have

$$S_j \neq 0 , \quad j = 0, 1, 2, 3, 4. \qquad (12)$$

Therefore for v = 1, equations (10) and (12) must be complied with simultaneously.

Lemma 1: Equations (10) and (12) are equivalent to

$$\begin{cases} \beta_0 = \beta_1 = \beta_2 = 0 & (13.1) \\ S_1 \neq 0 & (13.2) \end{cases}$$

Proof:

If (10) and (12) are true, then (13) must be true since the latter is a subset of the former.

Conversely, assume that (13.1) and (13.2) hold. $S_1 \neq 0$ and $\beta_0 = 0$ imply $S_0 \neq 0$ and $S_2 \neq 0$; $S_1 \neq 0$, $S_2 \neq 0$, *and* $\beta_1 = 0$ imply $S_3 \neq 0$; $S_1 \neq 0$, $S_2 \neq 0$, $S_3 \neq 0$, *and* $\beta_2 = 0$ imply $S_4 \neq 0$. That is, $S_j \neq 0$, j = 0, 1, 2, 3, 4. Using this fact, equation (13.1) can be rewritten as $S_1/S_0 = S_2/S_1$, $S_2/S_1 = S_3/S_2$, *and* $S_2/S_1 = S_4/S_3$ respectively. Therefore, equation (10) must be true.

Q.E.D.

Now consider the case of v = 2, i.e., there are two byte-errors in the received word. Then equation (7) becomes

$$\begin{cases} S_0\sigma_2 + S_1\sigma_1 + S_2 = 0 & (14.1) \\ S_1\sigma_2 + S_2\sigma_1 + S_3 = 0 & (14.2) \\ S_2\sigma_2 + S_3\sigma_1 + S_4 = 0 & (14.3) \end{cases}$$

and (6) reduces to

$$\sigma(x) = x^2 + \sigma_1 x + \sigma_2 \qquad (15)$$

The coefficients σ_1 *and* σ_2 can be obtained from (14.1) and (14.2) as

$$\sigma_1 = \begin{vmatrix} S_0 & S_2 \\ S_1 & S_3 \end{vmatrix} \Big/ \begin{vmatrix} S_0 & S_1 \\ S_1 & S_2 \end{vmatrix} = \frac{\beta_4}{\beta_0} \qquad (16.1)$$

$$\sigma_2 = \begin{vmatrix} S_2 & S_1 \\ S_3 & S_2 \end{vmatrix} \Big/ \begin{vmatrix} S_0 & S_1 \\ S_1 & S_2 \end{vmatrix} = \frac{\beta_1}{\beta_0} \qquad (16.2)$$

or equivalently from (14.2) and (14.3) as

$$\sigma_1 = \beta_2/\beta_1 \qquad (16.3)$$

$$\sigma_2 = \beta_3/\beta_1 . \qquad (16.4)$$

Since v = 2 is assumed, we have $\sigma_1 \neq 0$ *and* $\sigma_2 \neq 0$ in order for equation (15) to have two different roots. Using this fact and using equations (16.1) to (16.4), it follows that the conditions below must be conformed to simultaneously for v = 2,

$$\begin{cases} \beta_i \neq 0, \quad i = 0,1,2,3,4 & (17.1) \\ \beta_1\beta_4 = \beta_0\beta_2, \ \textit{and} \ \beta_1^2 = \beta_0\beta_3 & (17.2) \end{cases}$$

Lemma 2: Equations (17) is equivalent to

$$\begin{cases} q = S_0\beta_3 + S_1\beta_2 + S_2\beta_1 = 0 & (18.1) \\ \beta_0 \neq 0, \quad \beta_1 \neq 0, \quad \beta_2 \neq 0 . & (18.2) \end{cases}$$

Proof

If (17) holds, then $\beta_1\beta_4 = \beta_2\beta_0$, and $\beta_1^2 = \beta_0\beta_3$. After expanding and re-arranging them, we have the following equations.

$$S_1(S_2^3 + S_0S_3^2 + S_1^2S_4 + S_0S_2S_4) = 0,$$

and

$$S_2(S_2^3 + S_0S_3^2 + S_1^2S_4 + S_0S_2S_4) = 0.$$

Let

$$q = S_2^3 + S_0S_3^2 + S_1^2S_4 + S_0S_2S_4,$$

thus

$$S_1 q = S_2 q = 0.$$

If $q \neq 0$, then $S_1 = S_2 = 0$. Thus, $\beta_0 = \beta_1 = \beta_2 = 0$. This conflicts (17.1). As a result, we have $q = 0$. So if (17) holds, then (18) holds.

On the other hand, if (18) holds, it is easy to see that $\beta_1\beta_4 = \beta_2\beta_0$ and $\beta_1^2 = \beta_0\beta_3$. As $\beta_0 \neq 0$, $\beta_1 \neq 0$, and $\beta_2 \neq 0$, so that $\beta_3 \neq 0$, and $\beta_4 \neq 0$. So if (18) holds, then (17) holds.

Q.E.D.

So if two byte-errors occur, (18) must be satisfied.

In the case of v=3, It is easy to see the determinant of $q = \begin{vmatrix} S_0 & S_1 & S_2 \\ S_1 & S_2 & S_3 \\ S_2 & S_3 & S_4 \end{vmatrix}$ not equal to zero. This is because $\begin{vmatrix} S_0 & S_1 & S_2 \\ S_1 & S_2 & S_3 \\ S_2 & S_3 & S_4 \end{vmatrix}$ can be decomposed into a product of two error location Vandermonde matrixes with an error value matrix, that is

$$\begin{vmatrix} S_0 & S_1 & S_2 \\ S_1 & S_2 & S_3 \\ S_2 & S_3 & S_4 \end{vmatrix} = \begin{vmatrix} X_1^{-2} & X_2^{-2} & X_3^{-2} \\ X_1^{-1} & X_2^{-1} & X_3^{-1} \\ 1 & 1 & 1 \end{vmatrix} \cdot \begin{vmatrix} Y_1 & 0 & 0 \\ 0 & Y_2 & 0 \\ 0 & 0 & Y_3 \end{vmatrix} \cdot \begin{vmatrix} 1 & X_1 & X_1^2 \\ 1 & X_2 & X_2^2 \\ 1 & X_3 & X_3^2 \end{vmatrix} .$$

As long as v=3, none of X_1, X_2, X_3, Y_1, Y_2, and Y_3 will be zero. As a result, the determinant of q does not equal to zero . However, if (17) holds, the determinant of q must be zero. This can be showed as follows: define $\sigma_1 = \beta_2/\beta_1$ and $\sigma_2 = \beta_3/\beta_1$, so that σ_1 *and* σ_2 must be the solution of the following joint equation

$$\begin{cases} S_0\sigma_2 + S_1\sigma_1 + S_2 = 0 \\ S_1\sigma_2 + S_2\sigma_1 + S_3 = 0 \\ S_2\sigma_2 + S_3\sigma_1 + S_4 = 0 \end{cases}$$

Since this equation has only two variables, the determinant of q must be zero. Thus for the case of v=3, we can claim that (17), and also (18), will not be satisfied.

It is well known that (15) will have two different roots in $GF(2^m)$ if and only if $T_2(K) = \sum_{i=0}^{m-1} K^{2^i} = 0$ where

$K = \sigma_2 / \sigma_1^2$ and $T_2(K)$ is known as the trace of K.

Let W(S) and W(β) denote the Hamming weights of $S = (S_0 \ S_1 \ S_2 \ S_3 \ S_4)$ and $\beta = (\beta_0 \ \beta_1 \ \beta_2)$, respectively. From the above discussion, we arrive at the following decoding algorithm:

(1) If W(S) = 0, then set v = 0;
(2) If W(β) = 0 and $S_1 \neq 0$, then set v = 1;
(3) If W(β) = 3, q = 0, and $T_2(K) = 0$, then set v = 2;
(4) For cases other than the mentioned above, set v $\geq$ 3.

Fig. 1 is a flow chart of the above decoding algorithm for DBEC-TBED RS codes. The Koksal-Yucel's version of Deng-Costello's algorithm is displayed in Fig.2 for comparison. It is easy to see that the new algorithm is simpler and more efficient compared with Koksal-Yucel's one. Once the number of byte-errors v is found, the calculation of the error location numbers X_i and the error values Y_i is straight forward, just based on (9) and (10) for single error case and (15) for double error case.

III. EXAMPLE

The following example shows how the decoding technique stated above can detect the triple-byte errors that the Deng-Costello algorithm fails to detect. For comparison, this example is taken from [2] used by Koksal and Yucel. The DBEC-TBED RS code is a (15,10) RS code over $GF(2^4)$ with primitive polynomial $x^4 + x + 1 = 0$ and primitive element α. The generator polynomial of the code assuming $m_0 = -2$ in (1) is given by

$$g(x) = \prod_{i=-2}^{2} (x - \alpha^i) = x^5 + 3x^4 + Ex^3 + Ex^2 + 3x + 1$$

The code polynomial is

$$c(x) = 3 + 7x + 3x^2 + 9x^3 + 3x^4 + 9x^5 + 8x^6 + 5x^7 + 3x^8 + 3x^9 + Ax^{10} + 4x^{11} + 9x^{12} + Ex^{13}$$

The triple byte-error polynomial which corrupts the code polynomial c(x) is $e(x) = 9x^{12} + Fx^7 + x^6$. Thus, $S = (S_0 \ S_1 \ S_2 \ S_3 \ S_4) = (3 \ 8 \ 7 \ 1 \ 4)$, $\beta_0 = 5$, $\beta_1 = E$, $\beta_2 = 1$. Since the weight of β is 3, compute $\beta_3 = S_3^2 + S_2 S_4$ *and* $q = S_0\beta_3 + S_1\beta_2 + S_2\beta_1$ to obtain $\beta_3 = 8$ and $q = F$. Since $q \neq 0$, triple byte-errors are detected.

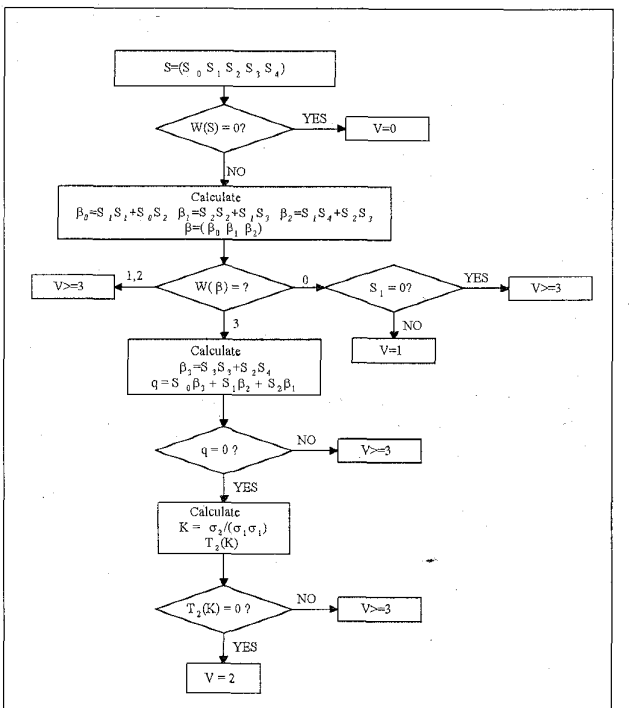

Figure 1 The new algorithm for DBEC-TBED RS Codes

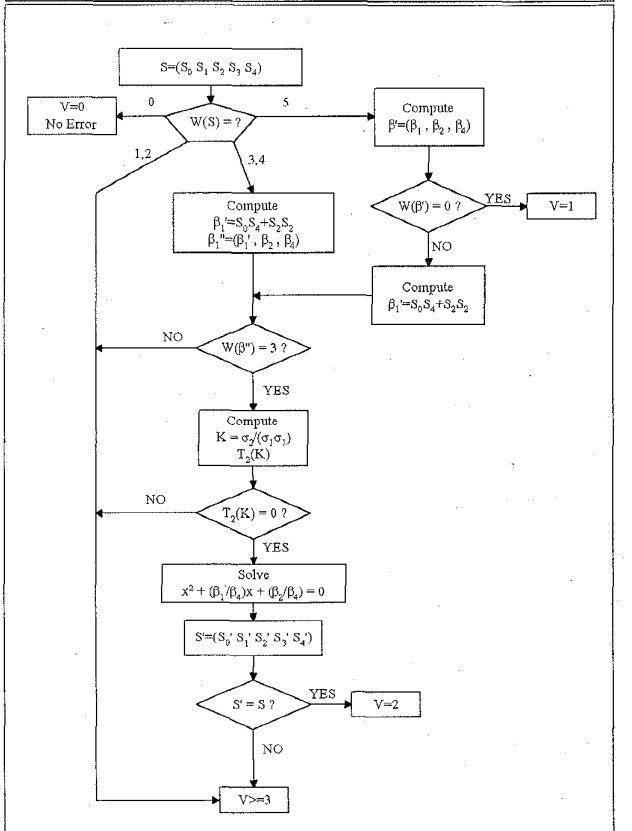

Figure 2 Koksal-Yucel's modified algorithm for decoding DBEC-TBED RS Co

REFERENCES

[1] R. H. Deng and D.J. Costello, Jr., " Decoding of DBEC-TBED Reed-Solomon codes, " IEEE Trans. Comput., vol. C-36, pp.1359-1363, Nov., 1987.

[2] F. Z. Koksal and M.D. Yucel, "Comments on the decoding algorithms of DBEC-TBED Reed-Solomon codes," IEEE Trans. Comput., C-41, pp. 244-247, 1992.

[3] W.W. Peterson and E.J. Weldon, Jr., Error-Correcting Codes, The MIT Press: England, 1961